Software Testing Assignment

# Module-2(Manual Testing)

--------------------------------------------------------------------------------------------------------------------------------------\* What is Exploratory Testing?

**-Though the current trend in testing is to push for automation,** exploratory testing is a new way of thinking. **Automation has its limits**

**-** Is not random testing but it is Ad-hoc testing with purpose of find bugs.

**-** Is structured and rigorous.

**-** Is highly teachable and manageable.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is traceability matrix?

**-** Test conditions should be able to be linked back to their sources in the test basis ,this is known as traceability.

**-** To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

Types of Traceability Matrix:-

**Forward Traceability-** Mapping of requirements to Test cases

**Backward Traceability-** Mapping of Test cases to requirements

**Bi-Directional Traceability-** A good traceability matrix is the references from test cases to basis documentation and vice versa.

Pros of Traceability Matrix:-

**.** To make sure that all requirements included in the test cases

**.** To make sure that developers are not creating features that no one has requested.

**.** Easy to identify the missing functionalities.

Cons of Traceability Matrix:-

**.** No traceability or Incomplete Traceability Results into:

**.** Poor or unknown test coverage, more defects found in production.

**.** Difficult project planning and tracking, misunderstanding between different teams over project dependencies, delays, etc.

# \* What is Boundary value testing?

**-** Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.

**-** Boundary value analysis is a method which refines equivalence partitioning.

**-** Boundary value analysis generates test cases that highlight error better than equivalence partitioning.

**-** Aim is to treat groups of input as equivalent and to select one representative input to test them all.

**-** Equivalence partitioning can be used for all levels of testing.

**-** Equivalence partitioning is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is integration testing?

**-** Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

**-** Integration testing is a level of the software testing process where individual units are combined and tested as a group.

**-** Integration testing is done by a specific integration tester or test team.

**-** The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in integration testing.

**- There are 2 levels of integration testing:-**

**.** Component integration testing

**.** System integration testing

--------------------------------------------------------------------------------------------------------------------------

# \* What determines the level of risk?

**-** A properly designed test that passes reduces the overall level of risk in a system.

**-** A factor that could result in future negative consequences; usually expressed as impact and likelihood.

**-** When testing does find defects, the Quality of the software system increases when those defects are fixed.

**-** The Quality of system can be improved through lessons learned from previous projects.

**Types of Risk:-**

**\* Project Risk \* Product Risk**

**-** A risk could be any future event with a negative consequence you need to identify the risks associated with your project.

**--------------------------------------------------------------------------------------------------------------------------------------**

# \* What is Alpha testing?

**-** It is always performed by the developer at the software development site.

**-** Sometimes it is also performing by independent testing team.

**-** Alpha testing is not open in the market and public.

**-** It is conducted for the software application and project.

**-** It is always performed in virtual environment.

**-** It is always performed within organization.

**-** It is the form of acceptance testing.

**-** It comes under category of both White box testing and Black box testing.

**--------------------------------------------------------------------------------------------------------------------------------------**

# \* What is beta testing?

**-** It is always performed by the customers at their own site.

**-** It is not performed by independent testing plan.

**-** Beta testing is always open to the market and public.

**-** It is usually conducted for software product.

**-** It is performed in real time environment.

**-** It is always perform outside the organization.

**-** It is also the form of acceptance testing.

**-** It is only a kind of black box testing.

**-** Beta testing can be considered “pre-releasing” testing.

**--------------------------------------------------------------------------------------------------------------------------------------**

# \* What is component testing?

**-** The testing of individual software component.

**-** Unit/component testing is a level of the software testing process where individual units/components of software are tested.

**-** Unit or component is the smallest testable part of the software.

**-** Unit testing or component testing is the first level of testing and is performed prior to integration testing.

**-** Sometimes known as unit testing, module testing, or program testing.

**-** Unit testing is performed by using the white box testing method.

system, Check the sanity/rationality, done by testers

**6) End to End:-**It checks the complete end-to-end process and flow features.

**-** Validates the software system as well as interconnected sub-systems

**-** Component can be tested in isolation –Stubs/Drivers may be employed.

------------------------------------------------------------------------------------------------------------------------------------

# \* What is functional system testing?

**-** Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.

**-** Testing based on an analysis of the specification of the functionality of a component or system.

**-** Functional testing is executed first

**-** Manual testing or automation tools can be used for functional testing

**-** Easy to do manual testing , Business requirement are the inputs to functional testing

**-** Function testing describes what the product does.

**-** Types:-

1) Black box 4) Smoke

2) White box 5) Sanity

3) Experience based testing 6) End to End

**1) Black box testing:-**Specification-based testing technique is also known as ‘Black-box’or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

**-** The testers have no knowledge of how the system or component is structured inside the box.

**2) White box testing:-**Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

**3) Experience based testing:-**In experience-based techniques, people’s knowledge, skills and background are of prime importance to the test conditions and test cases.

**-** Testers, users, stake holders are done.

**4) Sanity:-**It is done toCheck critical functionality, check the stability, Check the system end to end

and it is done in initial stage.

**5) Smoke:-**It is done to Check the new functionality, Check only a particular function of entire

---------------------------------------------------------------------------------------------------------------------------------

# \* What is Non-Functional testing?

**-** Non-Functional testing checks the performance, reliability, scalability, and other non-functional aspects of the software system.

**-** Non-Functional testing should be performed after functional testing.

**-** Using tools will be effective for this testing.

**-** Performance parameters like speed, scalability are inputs to non-functional testing.

**-** Tough to do manual testing

* **Types:-**
* Usability testing
* Compatibility testing
* GUI testing
* Security testing
* Performance testing
* Stress testing
* Load testing

# \* What is GUI testing?

**-** Graphical user interface (GUI) testing is the process of testing the system GUI of the system under test.

**-** GUI testing involves checking the screens with the control like menus, button, icons, and all types of bars like tool bar, menu bar, dialog boxes and windows etc.

\*Approach of GUI testing:-

* **MANUAL BASED TESTING** :- graphical screens are checked manually by testers in conformance with the requirement in business requirement documents.
* **RECORD AND REPLAY** :- GUI testing done using automation tools this is done in 2 parts. During Record, tests step are captured into the automation tool. During Playback, recorded test steps are executed on the application under test.
* **MODEL BASED TESTING**:- A model is a graphical description of system’s behaviour. It helps us to understand and predict the system behaviour models help in a generation of efficient test case using the system requirements.

------------------------------------------------------------------------------------------------------------------------------

# \* What is Ad-hoc testing?

**-** Ad-hoc testing is an informal testing type with an aim to break the system.

**-** Ad-hoc testing does not follow any structured way of testing and it is randomly done on any part of application

**-** Main aim of this testing is to find defects by random checking.

**-** Ad-hoc testing can be achieved with the testing technique called ‘Error Guessing’.

**-** No need to compare to the documents

**-** It is used after formal testing.

\* Types:-

* Buddy testing:- Developer and tester work together.
* Pair testing:- Tester and scriber, both are testers work together.
* Monkey testing:- Randomly test, experience based tester do testing.

-------------------------------------------------------------------------------------------------------------------------------

# \* What is load testing?

**-** Load testing is to test the system behaviour under normal workload conditions and it is just testing or simulating with the actual work load.

**-** Load testing does not break the system.

**-** Load testing identifies the bottlenecks breaking the system under various work load and check how the system reacts when the load is generally increased.

-------------------------------------------------------------------------------------------------------------------------------------

# \* What is stress testing?

**-** Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.

**-** In stress testing determines the point of the system to reveal the maximum point after which it breaks.

**-** In stress testing tries to break the system by testing with overwhelming data or resources.

Types of stress testing :-

* Application stress testing
* Transactional stress testing
* Systemic stress testing
* Exploratory stress testing

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is white box testing and list the types of white box testing?

**-** White box testing is the detailed investigation of internal logic and structure of the code.

**-** Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

**-** Testing based on an analysis of the internal structure of the component or system.

**-** Based on code and the design of the system

**TEST/CODE COVERAGE**

* Coverage = no. of coverage items/Total no. of coverage \*100%
* BENEFIT:
* It creates additional test cases to increase coverage.
* It helps in finding areas of a program not exercised by a set of test cases.
* DRAWBACK:
* One drawback of code coverage measurement is that it measuring coverage of what has been written. i.e. the code itself, it cannot say anything about the software that has not been written.

**Types of coverage :-**

* Statement coverage:- The statement coverage covers only the true conditions.
* Decision coverage:- it covers both the true and false conditions.
* Condition coverage:-it reports the true or false outcome of each conditions.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is black box testing? What are the different black box testing techniques?

**-** Specification-based testing technique is also known as ‘Black-box’or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

**-** The testers have no knowledge of how the system or component is structured inside the box.

INPUT

OUTPUT

If Output = Expected result then pass

**\*Advantages:-**

**-** Code access not required

**-** Well suited and efficient for large code segments.

**-** Clearly separates user’s perspective from the developer’s perspective through visibly defined roles.

**\*Disadvantages**

**-** Inefficient testing, due to the fact that the tester only has limited knowledge about an application.

**-** Blind coverage, since the tester cannot target specific code segments or error prone areas.

**-** The test cases are difficult to design.

* **Techniques/Types of Black box testing**
* **Equivalence partitioning**:- Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.
* **Boundary value analysis**:- B.V.A. is a method which refines equivalence partitioning
* **Decision table**:- Decision table is a good way to deal with combinations of thing
* **State transition testing**:- A black box test design technique in which test cases are designed to execute valid and invalid state transitions also known as N-switch testing.
* **Use-case testing**

--------------------------------------------------------------------------------------------------------------------------------------

# \* Mention what are the categories of defects?

* **Data Quality/Database Defects**: Deals with improper handling of data in the database.
* Critical Functionality Defects: The occurrence of these bugs hampers the crucial functionality of the application.
* **Functionality Defects**: These defects affect the functionality of the application
* **Security Defects**: Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
* **User Interface Defects**: As the name suggests, the bugs deal with problem related to UI are usually considered less severe.

-------------------------------------------------------------------------------------------------------------------------------------

# \* Mention what big bang testing is?

**-** In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as whole.

**-** Big Bang testing has the advantages that everything is finished before integration testing starts.

**\*** **Advantages**:

**-** Convenient for small systems.

**\*Disadvantages**:

**-** Fault localization is difficult.

**-** Some interface links to be tested could be missed easily.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is the purpose of exit criteria?

**-** Exit criteria is used to determine whether a given test activity has been completed or NOT.

**-** Exit criteria can be defined for all of the test activities right from planning, specification and execution.

**-** Exit criteria should be part of test plan and decided in the planning stage.

-------------------------------------------------------------------------------------------------------------------------------------

# \* When should “Regression Testing” be performed?

**-** Change in requirements and code is modified according to the requirement.

**-** New feature is added to the software

**-** Defect fixing

**-** Performance issue fix.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is 7 key principles? Explain in detail?

**- 1) Testing show presence of defects :** We test to find faults.

**- 2) Exhaustive testing is impossible :** Testing everything including all combinations of inputs and precondition is not possible.

**- 3) Early testing :** Testing activities should start as early as possible in the development life cycle.

**- 4) Defect clustering :** All small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failure.

**- 5) The pesticide paradox :** If the same tests are repeated over and over again, eventually the same set of test case will no longer find any new defects.

**- 6) Testing is context dependent :** Testing is basically context dependent. Testing is done differently in different contexts. Different kinds of sites are tested differently.

**- 7) Absence of error fallacy :** Even after defects have been resolved it may still be unusable and/or does not fulfil the user’s needs and expectations.

--------------------------------------------------------------------------------------------------------------------------------------

# \* Difference between QA v/s QC v/s Tester?

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Tester** |
| Process-oriented activities | Product-oriented activities | Product-oriented activities |
| Preventive activities | Corrective process | Preventive activities |
| QA is process related | QC is actual testing software |  |
| QA is entire life cycle | QC for testing part in SDLC |  |
| Focuses on building in quality | Focuses on testing for quality | Focuses on actual testing. |
| QA is subset of software test life cycle (STLC) | QC is can be considered as the subset of QA(Quality assurance) | Testing is subset of QC(Quality control) |

--------------------------------------------------------------------------------------------------------------------------------------

# \* Difference between Smoke and Sanity?

|  |  |
| --- | --- |
| **Smoke** | **Sanity** |
| Check Critical functionality | Check and verify the new-functionality |
| It checks the system end to end | It check only a particular function of entire system |
| It is done in initial stage | It is done after 30 build |
| It check the stability | It check the sanity/rationality |
| Part of acceptance testing | Part of regression testing |
| Done by tester and developer | Done by tester |

--------------------------------------------------------------------------------------------------------------------------------------

# \* Difference between verification and validation.

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Verification check whether we are building the right product | Validation check whether we are building the product right |
| Focus on document | Focus on actual software |
| Process right or not | Take place after verification are completed |

--------------------------------------------------------------------------------------------------------------------------------------

# \* Explain types of performance testing.

**-** Focus of performance testing is checking a speed, scalability, and stability.

* **Types of performance testing:-**
* **Load testing**
* **Stress testing**
* **Endurance testing**
* **Spike testing**
* **Volume testing**
* **Scalability testing**

**\* Performance problem:-**

**Load** **a** **long** **time**:- load time is normally the initial time it takes an application to start.

**Poor** **response** time:-Response time is the time it takes from when a user input data into the application until the application outputs a response to that input .

**Poor** **scalability**:- A software suffers from poor scalability when it cannot handle the expected number of user

**Bottlenecking**:- Bottlenecking is when either coding errors or hardware issues cause a decrease of through put under certain loads.

**\* Performance testing tools:-**

HP load runner

HTTP load

Proxy sniffer

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is Error, Defect, Bug and failure?

* **Error**:- A mistake in coding OR Any abnormal behaviour is known as Error e.g. spelling mistake, wrong syntax etc.
* **Defect**:- When the software does not behave as per user need OR Error found by tester is known as defect.
* **Bug**:- IF the developer agrees that this is actually a defect then it is bug.
* **Failure**:- If the software system is not as per SRS(software requirement system) OR build does not meet requirements then it is failure.

--------------------------------------------------------------------------------------------------------------------------------------

# \* Difference between priority and severity.

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority describes the importance of the defects | Severity describes the seriousness of defect and how much impact on business work flow |
| Priority is relative and business-focused .Priority defines the order in which we should resolve a defect. | Severity is absolute and customer-focused. It is the extent to which the defect can affect the software. |
| The priority status is set based on customer requirements. |  |
| Priority can be following types:-  Critical, P0(High), P1(Medium), P2(low) | Severity can be following types:-  Critical, Major(High), Moderate(Medium), Minor(Low) |
| Developer can fix it in later releases | It won’t cause any major break down of the system |

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is Bug Life Cycle?

**-** The duration or time span between the first time defect is found and the time that it is closed successfully, rejected, postponed, or deferred is called as ‘Defect Life Cycle’

* **Bug(Defect) Life Cycle:-**

NEW

ASSIGNED

DUPLICATE  
REJECTED  
DEFFERED  
NOT A BUG

OPEN

FIXED

RE-OPENED

CLOSED

VERIFIED

RETEST

PENDING RETEST

# \*Explain the difference between Functional testing and Non-functional testing.

|  |  |
| --- | --- |
| **Functional testing** | **Non-functional testing** |
| Functional testing is executed first. | Non-functional testing should be performed after functional testing. |
| Manual and automation testing tools can be used for functional testing. | Using tools will be effective for this testing. |
| Functional testing describes what the product does. | Non-Functional testing describe how software works. |
| Easy to do manual testing. | Tough to do manual testing. |
| Concentrate on user requirement. | Concentrate on user expectation. |
| Business requirement are the input to functional testing. | Performance parameter like speed, scalability, are the input to non-functional testing. |
| Type of functional testing:-  Black box testing, White box testing, Experience testing, Smoke, Sanity, End to End. | Types of Non-functional testing:-  Usability testing, Compatibility testing, GUI testing, Security testing, Performance testing, Stress testing, Load testing. |

-------------------------------------------------------------------------------------------------------------------------------------

# \*What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle) .

|  |  |
| --- | --- |
| **STLC** | **SDLC** |
| STLC is mainly related to software testing. | SDLC is mainly relates to software development. |
| It focuses only on testing the software. | Besides development other phases like testing is also included. |
| On only testing process. | On both the development and testing process. |
| In STLC, less number of members (testers) are needed. | In SDLC, more number of members (developers) are required for the whole process. |
| In STLC testing team (Test Lead or Test Architect) makes the plans and designs | In SDLC, the development team makes the plans and designs based on the requirements |
| A bug free software | A good quality software product |
| It helps in making the software defect free. | It helps in developing good quality software. |
| STLC phases are performed after SDLC phases. and It is Testing life cycle | SDLC phases are completed before STLC phases. And It is Development life cycle |

--------------------------------------------------------------------------------------------------------------------------------------

# \*What is the difference between test scenarios, test cases, and test script?

* **Test scenarios:** Test scenario is ‘What to be tested’. Test scenario is nothing but test procedure.
* **Test cases:** Test cases is ‘How to be tested’ . Test cases are derived or written from test scenario.
* **Test script:** A test script in software testing is a set of instruction that will be performed on the system under test to test that system function as expected.

**\*** There are various means for executing test scripts.

**.** Manual Testing

**.** Automation Testing

--------------------------------------------------------------------------------------------------------------------------------

# \*Explain what Test Plan is? What is the information that should be covered .

**-** A test plan is document that describe the test scope, test strategy, objectives, deliverables and resources requirement

**-** The test plan strategy along with efforts and cost estimates for the project.

**-** Moreover, the resources, test environment, test limitations and the testing schedule are also determined.

**-** The test plan gets prepared and finalized in the same phase.

**-** Below information should be covered:-

Overview, Scope, Inclusions, Test environment, Exclusions, Test strategy, Defect reporting procedure, Roles/responsibilities, Test schedule, pricing, Tools, Approvals, Risk and mitigations, Test deliverables, Test tools selections, Test formations, Test effort estimation.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is Priority?

|  |
| --- |
| Priority describes the importance of the defects |
| Priority is relative and business-focused .Priority defines the order in which we should resolve a defect. |
| The priority status is set based on customer requirements. |
| Priority can be following types:-  Critical, P0(High), P1(Medium), P2(low) |
| Developer can fix it in later releases |

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is Severity?

|  |
| --- |
| Severity describes the seriousness of defect and how much impact on business work flow |
| Severity is absolute and customer-focused. It is the extent to which the defect can affect the software. |
| Severity can be following types:-  Critical, Major(High), Moderate(Medium), Minor(Low) |
| It won’t cause any major break down of the system |

--------------------------------------------------------------------------------------------------------------------------------------

# \* Bug categories are…

* **Duplicate**: If the defect is repeated twice or the defect corresponds to the same concepts of the bug is called duplicate.
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected”.
* **Deferred**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release it called deffered.
* **Not a bug**: If it does not affect the functionality of the application then it called not a bug.

--------------------------------------------------------------------------------------------------------------------------------------

# \* Advantage of Bugzila .

* Open source, free bug tracking tool.
* Automatic Duplicate Bug Detection.
* Search option with advanced features.
* File/Modify Bugs By Email.
* Move Bugs Between Installs.
* Multiple Authentication Methods (LDAP, Apache server).
* Time Tracking.
* Automated bug reporting; has an API to interact with system

--------------------------------------------------------------------------------------------------------------------------------------

# \* What are the different methodologies in Agile Development Model?

* **Customer satisfaction through early and continuous software delivery:**
* **Accommodate changing requirements throughout the development process:**
* **Frequent delivery of working software:**
* **Collaboration between the business stakeholders and developers throughout the project:**
* **Support, trust, and motivate the people involved:**
* **Enable face-to-face interactions:**
* **Working software is the primary measure of progress:**
* **Agile processes to support a consistent development pace:**
* **Attention to technical detail and design enhances agility**
* **Simplicity:**
* **Self-organizing teams encourage great architectures, requirements, and designs:**
* **Regular reflections on how to become more effective:**

---------------------------------------------------------------------------------------------------------------------------

# \* Explain the difference between Authorization and Authentication in web testing. What are the common problems faced in web testing?

* **Authentication**: Accepting an invalid username/password
* **Authorization**: Accessibility to pages through permission not given

--------------------------------------------------------------------------------------------------------------------------------------

# \* When to used Usability testing?

**-** Usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you’ve begun the strategy work around a brand new site or app.

**-** This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants. Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible.

--------------------------------------------------------------------------------------------------------------------------------------

# \* What is the procedure for GUI Testing?

**-** Graphical user interface (GUI) testing is the process of testing the system GUI of the system under test.

**-** GUI testing involves checking the screens with the control like menus, button, icons, and all types of bars like tool bar, menu bar, dialog boxes and windows etc.

\*Approach of GUI testing:-

* **MANUAL BASED TESTING** :- graphical screens are checked manually by testers in conformance with the requirement in business requirement documents.
* **RECORD AND REPLAY** :- GUI testing done using automation tools this is done in 2 parts. During Record, tests step are captured into the automation tool. During Playback, recorded test steps are executed on the application under test.
* **MODEL BASED TESTING**:- A model is a graphical description of system’s behaviour. It helps us to understand and predict the system behaviour models help in a generation of efficient test case using the system requirements.

-------------------------------------------------------------------------------------------------------------------------------

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*